Inteligência Artificial Aplicada a Detecção de Fake News

Leandro Massetti R. Oliveira[1], **Aline Mayara S. Costa**[1], **Vandecia Rejane M. Fernandes**[1]

[1]Universidade Federal do Maranhão (UFMA)
São Luís - MA - Brasil

massetti.leo@gmail.com, acostta2@gmail.com, vandecia@ecp.ufma.br

Abstract. *Information conveyed by the media does not always present with reliable content. Fake News is news that does not represent reality, but is shared as truthful. The malicious use of this information can compromise society by manipulating the opinion of readers. This work aims to do automatic detection of Fake News through Artificial Intelligence techniques, using algorithms such as Decision Tree, Naive Bayes, SVM and Deep Learning. Two approaches are used, obtaining an accuracy of 92.36 % in the best case.*

Resumo. *Informações veiculadas pela mídia nem sempre se apresentam com conteúdo confiável. As Fake News são notícias que não representam a realidade, mas que são compartilhadas como se fossem verídicas. O uso mal intencionado dessas informações pode comprometer a sociedade, manipulando a opinião do leitor. Este trabalho se propõe a fazer a detecção automática de Fake News por meio de técnicas de Inteligência Artificial, utilizando algoritmos como Árvore de Decisão, Naive Bayes, SVM e Deep Learning. Utiliza-se duas abordagens, obtendo uma acurácia de 92,36% no melhor caso.*

1. Introdução

Notícia é a difusão de informação através da imprensa falada ou escrita [Aurélio 1986], é um acontecimento novo ou que divulga novidade sobre uma situação já existente. Essa pode ser repassada por fontes oficiais na qual a imparcialidade garante sua apresentação livre de opiniões ou adjetivos que as engrandeça ou as diminua perante a realidade dos fatos. No âmbito jornalístico, o conjunto de normas éticas que regem a atividade do jornalista garante que informações sejam repassadas sem alterações [Alberto 1970]. Existem também as notícias que são criadas com o intuito de manipular o leitor. Essas notícias, de acordo com [Tandoc Jr et al. 2018], são denominadas *Fake News* e sua identificação é necessária para sanar os prejuízos que acarretam.

É possível fazer uma verificação em fontes confiáveis quanto a credibilidade do assunto, porém é um processo trabalhoso e que de acordo com alguns estudos não vem demonstrando resultados [Monteiro et al. 2018, George and Keane 2006]. No entanto, existem características nas *Fake News* quanto ao seu padrão de escrita que as diferenciam de notícias verdadeiras, o que pode ser uma ferramenta em sua identificação. O uso de técnicas computacionais para lidar com notícias falsas, como o Processamento de Linguagem Natural (PLN), é relativamente recente [Rubin et al. 2016, Appling et al. 2015, Gimenes et al. 2017, Pérez-Rosas and Mihalcea 2015].

Este trabalho objetiva a identificação automática de *Fake News* dentre notícias reais utilizando técnicas de inteligência artificial e está organizado da seguinte forma: na Seção 2 é apresentada brevemente o conceito de *Fake News*; na Seção 3 é apresentada a metodologia desenvolvida; na Seção 4 os resultados obtidos e na Seção 5 é apresentada as considerações finais.

2. *Fake News* e Suas Características

As *Fake News* são notícias com conteúdo falso e sem embasamento, criadas com o intuito de enganar e manipular o leitor [Tandoc Jr et al. 2018]. Elas atingem de maneira negativa a sociedade, resultando em situações alarmantes que vão de pequenos cenários a esferas mundiais. O trabalho de [Granik and Mesyura 2017] destaca a semelhança entre *Fake News* e *spam*, que é o uso de sistemas de mensagens eletrônicas para enviar uma mensagem não solicitada [Wikipédia 2019]. Dentre essas mensagens, os que se destacam são os inúmeros erros gramaticais, a tentativa de afetar a opinião do leitor, o conjunto limitado de palavras semelhantes no texto, além de muita poluição visual e conteúdo(s) falso(s). Essas características presentes no *spam* permitem a utilização de abordagens semelhantes para a detecção de *Fake News*.

O trabalho de [Monteiro et al. 2018] apresenta o que aparenta ser o primeiro *corpus* de *Fake News* em português além de terem executado alguns testes com SVM sobre os dados obtendo até 89% de acurácia. Os autores [Granik and Mesyura 2017] apresentaram uma abordagem simples de detecção de *Fake News* utilizando o algoritmo de classificação *Naive Bayes*, eles alcançaram em seu trabalho uma acurácia de 74% nos dados de teste na classificação de *posts* da plataforma *Facebook*. Já [Gilda 2017] apresenta o uso de duas técnicas para identificação de *Fake News*, o *term frequency-inverse document frequency* (TF-IDF) de *bi-grams* e *probabilistic context free grammar* (PCFG), eles obtiveram como melhor resultado a acurácia de 77,2% com a técnica TF-IDF utilizando vários algoritmos de aprendizado de máquina.

3. Metodologia

Este trabalho utiliza duas abordagens para a detecção das *fake news*: a primeira por meio de algoritmos tradicionais de aprendizado de máquina e a segunda por meio do uso de *Deep Learning*. Todo o processo é explicado nas seções seguintes.

3.1. Obtenção do Conjunto de Dados

Para realizar o treinamento e teste dos algoritmos de aprendizagem de máquina é necessário uma base da dados com notícias reais e notícias falsas. Neste trabalho foi utilizado o "Fake.Br Corpus", que é um conjunto de dados de notícias verdadeiras e falsas criado por [Monteiro et al. 2018]. Este conjunto de dados possui 3600 notícias reais e 3600 notícias falsas. Essas notícias estão alinhadas por assunto e por tamanho para evitar o viés sobre os dados. Todos os detalhes da obtenção do conjuntos de dados pode ser visto no trabalho dos autores.

3.2. Pré-Processamento

A fase de pré-processamento realizada nas notícias foi feita utilizando a linguagem de programação Python 3.6 por meio da biblioteca de tratamento de texto NLTK (*Natural Language Toolkit*) [Bird et al. 2009] que oferece uma variedade de técnicas de processamento de linguagem natural. Desta forma, o pré-processamento se deu nos seguintes passos:

1. **Padronização dos termos**: Nessa etapa é na qual as palavras de todas as notícias são padronizadas por meio da eliminação de pontuação, eliminação de acentos, caracteres especiais e uniformização das letras, passando todas para caixa baixa.

2. **Remoção de *StopWords***: Nessa etapa são removidas as *stopwords*, que são palavras consideradas irrelevantes ao texto, pois não tem sentido próprio quando consideradas sozinhas. Como exemplos temos os artigos e as preposições.

3. ***Stemming***: *Stemming* é o processo de reduzir palavras flexionadas ao seu radical (*stem*), dessa forma as palavras flexionadas (como plural, sufixo temporal, formas de gerúndio) são reduzidas ao seu radical.

Com este processo, as notícias foram organizadas em forma de listas, na qual cada lista era uma notícia e cada elemento da lista era uma palavra reduzida ao seu radical (que não foi eliminada nas etapas 1 e 2).

3.3. Vetores de Características

Para criação dos vetores de características das notícias criou-se primeiramente uma *bag of words* com todas as palavras que ocorreram no conjunto de treinamento. A *Bag Of Words* deste trabalho contém palavras que aparecem no conjunto de textos com frequência mínima de três vezes. Em seguida utilizou-se duas abordagens, a primeira para utilização dos métodos tradicionais de aprendizado de máquina e a segunda para o treinamento da *Deep Learning*. As subseções a seguir definem os passos de cada abordagem.

3.3.1. Abordagem A

O vetor de características da abordagem A de cada notícia é formado utilizando a técnica *Term Frequency - Inverse Document Frequency* (TF-IDF). Essa técnica consegue avaliar a importância de uma palavra contida em um texto e sua relevância em todo o conjunto de treino. Inicialmente, é calculada a frequência que uma palavra p_i aparece em uma notícia x dado por $TF(p_i|x)$. Em seguida, é calculada a frequência que essa mesma palavra aparece em todo o conjunto de dados $IDF = (p_i)$. Por fim, o cálculo do TF-IDF se dá pela Equação 1.

$$TF - IDF(x|p_i) = TF(x|p_i) \times \log_{10}(IDF(p_i)) \tag{1}$$

Com esse método, um exemplo de como ficaria a matriz de atributos pode ser vista na Tabela 1 na qual os dados da matriz foram normalizados. Desta forma, cada notícia corresponderá a uma linha na matriz de atributos e as palavras na *bag of words* corresponderão às colunas.

Tabela 1. Exemplo da matriz de atributos da Abordagem A.

'expuls'	**'kat'**	**'abr'**	**'pmdb'**	**'legend'**	...	**Classificação**
1	1	0,815625	0,76111111	0,525	...	Fake
0,38095238	0,15873016	0,51785714	0,36243386	1	...	Real
0,09756098	0	0	0	0	...	Fake
0	0	0	0	0	...	Real

3.3.2. Abordagem B

O vetor de característica na abordagem B é diferente, pois utiliza-se a técnica de *Word Embedding*. Com isso, o vetor de característica de cada notícia é caracterizado por um conjunto de inteiros representando o índice de cada palavra da notícia na *bag of words*. Na Tabela 2 é apresentado exemplos da representação das notícias desta forma. Nela é possível notar que a ordem das palavras nas notícias é preservada, pois há apenas a substituição de cada uma pelo seu índice na *bag of words*. Esse tipo de informação foi perdida utilizando a Abordagem A.

Tabela 2. Exemplos do vetor de característica da Abordagem B.

Notícia	Vetor de Característica	Classificação
1	$[2, 3, 12, 28, 206, 1, \cdots]$	Fake
2	$[19, 268, 1, 138, 64, 524, \cdots]$	Real
3	$[0, 731, 1120, 38, 9, 10, \cdots]$	Fake
4	$[98, 693, 43, 50, 65, 97, \cdots]$	Real

3.4. Treinamento

Feita a montagem das matrizes de atributos inicia-se a fase de treinamento. O treinamento foi feito utilizando as bibliotecas para Python *Scikit-Learn* [Pedregosa et al. 2011] para os algoritmos tradicionais e *TensorFlow* [TensorFlow: library 2019] para os algoritmos de *Deep Learning*. Dessa forma, foi necessário dividir os dados para treinamento e teste dos algoritmos. Como foram feitas duas abordagens, para cada uma tem-se um tipo de divisão de dados. A Abordagem A utilizou-se para os algoritmos clássicos de aprendizagem de máquina, enquanto que a Abordagem B utilizou-se para o treinamento das redes neurais. Para o treinamento das redes neurais, dividiu-se os dados em 80% para treinamento, 10% para validação e 10% para teste como é apresentado na Tabela 3

Tabela 3. Divisão do conjunto de dados Fake.Br.

Treinamento	5760
Validação	720
Teste	720
Total	7200

Os primeiros experimentos foram com os algoritmos tradicionais (SVM, Árvore de Decisão, *Naive Bayes*) nos quais foram realizados os treinamentos utilizando a matriz de atributos gerados pela Abordagem A no conjunto de treinamento junto com o conjunto de validação descrito na Tabela 3. Após o treinamento foi realizado a predição no conjunto de teste da mesma Tabela.

Com a matriz da Abordagem B foram realizados os treinamento das redes neurais. A escolha da arquitetura é um processo árduo. Utilizou-se uma arquitetura genérica, realizando o treinamento de várias redes e verificando-se o desempenho de cada uma nos dados de validação, almejando escolher as três arquiteturas com melhor desempenho. A Figura 1 representa a arquitetura genérica das redes utilizadas. Nessa arquitetura, a camada de *Embedding* visa representar as palavras em um vetor de dimensão previamente definida, e a camada LSTM *(Long Short Term Memory)* é necessária para a rede aprender a co-ocorrência das palavras nas notícias.

Seguindo o fluxograma, foram escolhidas 2 tipos de camadas *Embedding*, 3 opções de LSTM, 1 ou nenhuma camada densa e três opções quanto a quantidade de nós em cada camada. As opções escolhidas para as arquiteturas estão ilustradas na Tabela 4 na qual a combinação de cada uma gerou 36 arquiteturas. Entre cada etapa da Figura 1 foi inserido um *dropout* de 0.2 para evitar o *overfitting* ao conjunto de treinamento.

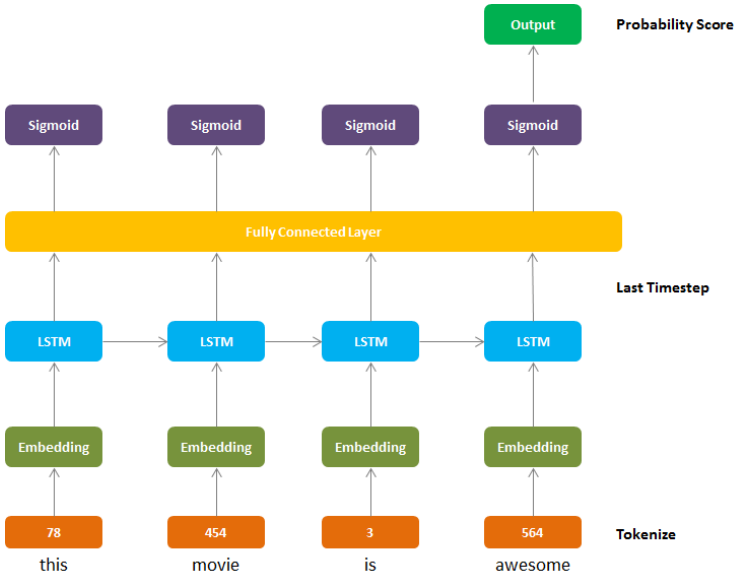

Figura 1. Arquitetura genérica das redes neurais treinadas. Fonte: [Samarth Agrawal, Towards Data Science 2019].

Tabela 4. Características das redes criadas.

Característica da Arquitetura	Opções
Dimensão do vetor *Embedding*	[64, 128]
Dimensão das camadas LSTM e densas	[32, 64, 128]
Quantidade de camadas LSTM	[1, 2, 3]
Quantidade de camadas densas	[0, 1]
Total	**36 arquiteturas**

Feito esse processo, deseja-se escolher as três melhores arquiteturas de acordo com a acurácia e custo de cada uma em suas execuções. Cada arquitetura foi treinada com o conjunto de dados por 80 épocas, que foi um tempo computacional razoável e não houve melhoras significativas na acurácia. Escolhida as três arquiteturas, fez-se a predição no conjunto de teste para comparação com os outros algoritmos.

4. Resultados

O resultado inicial das execuções é dado utilizando a Abordagem A descrita na Seção 3.3.2 na qual foram utilizados os algoritmos de classificação Árvore de Decisão, *Naive Bayes* e SVM. Foi utilizado o algoritmo *GridSearchCV* da biblioteca *scikit-learn* para obtenção do melhor valor para os parâmetros que utiliza o método *K-Fold Cross Validation* com $k = 3$ (Valor padrão do algoritmo *scikit-learn*) para cada combinação de parâmetros. Na Tabela 5 é apresentado o resultado inicial da execução para os dados de Teste.

Tabela 5. Resultados utilizando a Abordagem A.

Algoritmo	Acurácia	Precisão	Recall	Especificidade
Árvore de Decisão	77,77%	0,7793	0,775	0,7805
Naive Bayes	71,25%	0,7297	0,675	0,75
SVM	**90,13%**	**0,8753**	**0,9361**	**0,8667**

Do resultado dos algoritmos clássicos, o algoritmo que obteve melhor desempenho foi o SVM, que conseguiu 90% de acurácia nos dados de teste, enquanto que os outros dois se mantiveram na faixa dos 70% . É possível notar também que o SVM teve desempenho superior em todos as outras métricas, demonstrando que conseguiu definir bem um hiperplano separador dos dados. O resultado da execução do SVM confirmou os resultados obtidos por [Monteiro et al. 2018] com valores semelhantes de acurácia, que obteve 89% em seu trabalho.

Para a obtenção dos resultados utilizando a Abordagem B foram treinadas as 36 arquiteturas de redes descritas na Tabela 4. As redes foram treinadas com o conjunto de treino e testadas com o conjunto de validação. Nas Figuras 2 e 3 é apresentada a acurácia e o custo das redes nos dados de validação, onde em cada linha da Figura é representado o desempenho de cada uma das 36 redes.

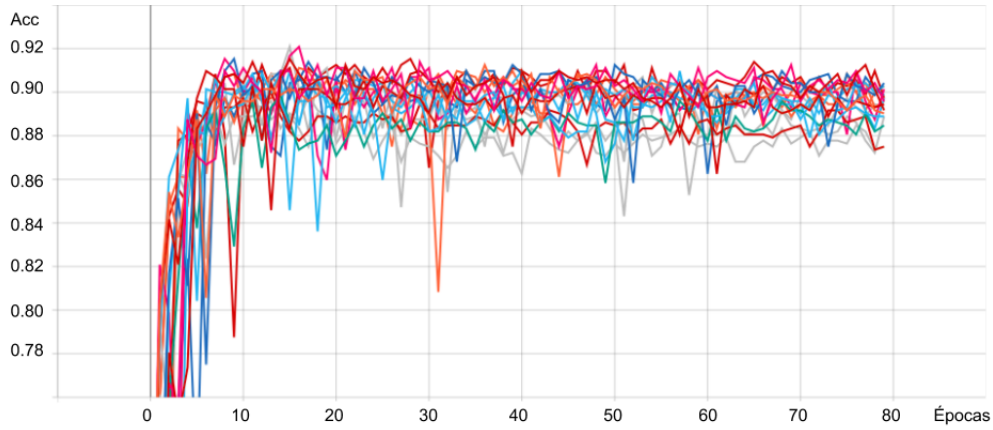

Figura 2. Acurácia das redes no conjunto de Validação.

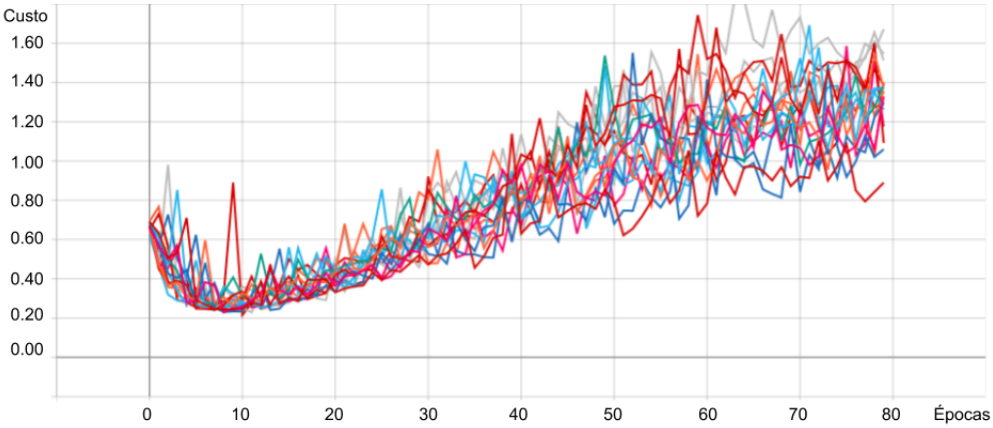

Figura 3. Custo das redes no conjunto de Validação.

É possível notar que para os dados de validação houve uma variação na acurácia das redes embora seja possível notar a tendencia global das mesmas em convergir para 90%. No entanto para os resultados de custo no conjunto de validação é possível notar um comportamento no aprendizado, inicialmente há uma queda, e depois os valores voltam a subir, demonstrando que as redes estão super-ajustando ao conjunto de treinamento.

Observando os valores, buscou-se as três melhores redes que tivessem a melhor relação acurácia-custo, deseja-se que as redes estejam no vale do gráfico de custo da validação e, ao mesmo

tempo, antes da estabilização dos valores de acurácia. A busca foi feita manualmente observando os valores de acurácia por época de treinamento a fim de escolher as três melhores redes. Na Tabela 6 é apresentada as três redes que tiveram o melhor desempenho com suas respectivas arquiteturas descritas na Tabela 4 e suas épocas.

Tabela 6. Arquitetura e desempenho das três melhores redes.

Rede	*Embedding*	Dimensão	LSTM	Dense	Época	Acurácia	Custo
1	128	64	3	1	11	90,14%	**0,2156**
2	128	128	1	1	9	90,97%	0,2299
3	128	128	3	1	16	**92,08%**	0,2785

Obtida as três redes, o próximo passo é verificar o seu desempenho no conjunto de dados de Teste para comparação com os resultados da Tabela 5. Na Tabela 7 é apresentado o desempenho das três redes nos dados de Teste.

Tabela 7. Resultados utilizando a Abordagem B.

Algoritmo	Acurácia	Precisão	Recall	Especificidade
Rede 1	**92,36%**	**0,9446**	0,9	**0,9472**
Rede 2	90,28%	0,8940	**0,9139**	0,8917
Rede 3	91,80%	0,9337	0,9	0,9361

É possível observar que a rede 1 se sobressaiu nos resultados atingindo a acurácia de 92,36% enquanto que as redes 1 e 2 se atingiram 90,28% e 91,80% respectivamente, e assim, obtiveram valores um pouco mais altos que o obtido pelo SVM na Tabela 5. Foi possível observar uma leve melhoria dos algoritmos de DL em comparação com o SVM, demonstrando que seus nós conseguiram generalizar bem obtendo os padrões das *Fake News*.

Considerando que é importante ser mais preciso sobre o que é *Fake News* e não marcar incorretamente uma notícia verdadeira como falsa, é necessário observar o valor da precisão. Quanto menor a precisão, significa que o classificador marcou mais notícias reais como *Fake News*.

Levando isso em consideração, a rede neural 1 teve um bom desempenho, pois em 94,46% das vezes que marcou algo como *Fake News*, ela acertou, enquanto que o SVM teve alto valor de *recall* demonstrando que acertou bem os exemplos da classe *Fake News* porém o baixo valor de especificidade evidencia que muitas notícias reais foram marcadas como falsas, o que não é desejável.

5. Conclusão

Os resultados da abordagem A mostram que o SVM se sobressaiu entre os demais obtendo acurácia de 90,13% enquanto Árvore de Decisão e *Naive Bayes* obtiveram acurácia abaixo dos 80%, com 77,77% e 71,25% respectivamente, além disso obtiveram um *recall* bem distante do SVM, constituindo valores abaixo de 0,7000. Esses último resultado, em especial, mostra que os algoritmos Árvore de Decisão, *Naive Bayes* apresentaram dificuldade de predizer quais dados realmente eram *Fake News* (verdadeiro-positivos).

Com a abordagem B foi possível encontrar uma rede que obteve desempenho de 92,36% de acurácia e 94,46% de precisão, apresentando uma melhoria em comparação com os outros

resultados. Essa melhoria é mais evidente quando se compara os valores de precisão e especificidade, dos quais as redes demonstraram não predizer erroneamente notícias reais como falsas, fato que é mais desejável para o problema de detecção de *Fake News*.

Referências

Alberto, A. (1970). *Ética e códigos da comunicação social*, volume 10. Editora SULINA.

Appling, D. S., Briscoe, E. J., and Hutto, C. J. (2015). Discriminative models for predicting deception strategies. In *Proceedings of the 24th International Conference on World Wide Web*, pages 947–952. ACM.

Aurélio, B. (1986). *Novo dicionário da língua portuguesa*. Nova Fronteira.

Bird, S., Klein, E., and Loper, E. (2009). *Natural language processing with Python*. "O'Reilly Media, Inc.".

George, J. and Keane, B. (2006). Deception detection by third party observers. In *deception detection symposium, 39th annual Hawaii international conference on system sciences*.

Gilda, S. (2017). Evaluating machine learning algorithms for fake news detection. In *2017 IEEE 15th Student Conference on Research and Development (SCOReD)*, pages 110–115. IEEE.

Gimenes, G., Cordeiro, R. L., and Rodrigues-Jr, J. F. (2017). Orfel: efficient detection of defamation or illegitimate promotion in online recommendation. *Information Sciences*, 379:274–287.

Granik, M. and Mesyura, V. (2017). *Fake news detection using naive Bayes classifier*.

Monteiro, R. A., Santos, R. L. S., Pardo, T. A. S., de Almeida, T. A., Ruiz, E. E. S., and Vale, O. A. (2018). *Contributions to the Study of Fake News in Portuguese: New Corpus and Automatic Detection Results*. Springer International Publishing.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Pérez-Rosas, V. and Mihalcea, R. (2015). Experiments in open domain deception detection. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1120–1125.

Rubin, V., Conroy, N., Chen, Y., and Cornwell, S. (2016). Fake news or truth? using satirical cues to detect potentially misleading news. In *Proceedings of the second workshop on computational approaches to deception detection*, pages 7–17.

Samarth Agrawal, Towards Data Science (2019). Sentiment Analysis using LSTM (Step-by-Step Tutorial) - Using PyTorch framework for Deep Learning.

Tandoc Jr, E. C., Lim, Z. W., and Ling, R. (2018). Defining "fake news" a typology of scholarly definitions. *Digital journalism*, 6(2):137–153.

TensorFlow: library (2019). TensorFlow.

Wikipédia (2019). Spamming - Wikipedia, the free encyclopedia.